1 Objectives

Prolog's greatest strength is its ability to backtrack to find alternative solutions. If not controlled, it can also be its greatest weakness. In this lecture we will go over the cut operator, which gives a solution to this problem.

- Know what the cut operator is and what it does.
- Know how to use the cut operator to assert failure.
- Know how to use the cut operator to stop recursion.

2 Examples

```
color(red).
color(blue).
car(honda).
car(ford).
car(toyota).

?- color(A), car(B).

A = red
B = honda ;
A = red
B = ford ;
A = red
B = toyota ;
A = blue
B = honda ;
A = blue
B = ford ;
A = blue
B = toyota ;
```

- The Cut operator (`!`) stops backtracking.
- It is considered a goal that always succeeds.

```
?- color(A), !, car(B).

A = red
B = honda ;
A = red
B = ford ;
A = red
B = toyota ;
No
```

3 Problems

Try the following problems. In a few minutes the instructor will go over the solutions. Feel free to work with the person next to you!

1. Write a predicate `between(X,Y,Z)` which is true when `Y` is a point between `X` and `Z`.
2. Write a predicate `grandfatherof(X,Y)` which is true when `X` is a grandfather of `Y`. You might want to write another predicate first.
3. Write a predicate `flatten(X,Y)` that is true when `Y` is a flattened version of `X`. E.g.

```
?- flatten([x,3,[a,b],3,[[3]]],[x,3,a,b,3,3]).
Yes
```

   Hint: there is a predicate called `is_list`, which is true when its argument is a list.